Package: SimTOST (via r-universe)

February 19, 2025

Title Sample Size Estimation for Bio-Equivalence Trials Through Simulation

Version 1.0.2

Description Sample size estimation for bio-equivalence trials is supported through a simulation-based approach that extends the Two One-Sided Tests (TOST) procedure. The methodology provides flexibility in hypothesis testing, accommodates multiple treatment comparisons, and accounts for correlated endpoints. Users can model complex trial scenarios, including parallel and crossover designs, intra-subject variability, and different equivalence margins. Monte Carlo simulations enable accurate estimation of power and type I error rates, ensuring well-calibrated study designs. The statistical framework builds on established methods for equivalence testing and multiple hypothesis testing in bio-equivalence studies, as described in Schuirmann (1987) <doi:10.1007/BF01068419>, Mielke et al. (2018) <doi:10.1080/19466315.2017.1371071>, Shieh (2022) <doi:10.1371/journal.pone.0269128>, and Sozu et al. (2015) <doi:10.1007/978-3-319-22005-5>. Comprehensive documentation and vignettes guide users through implementation and interpretation of results.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Imports MASS, Rcpp (>= 1.0.13), data.table, matrixcalc, parallel

Suggests PowerTOST, ggplot2, kableExtra, knitr, rmarkdown, testthat (>= 3.0.0), tibble, tinytest

VignetteBuilder knitr

Config/testthat/edition 3

LinkingTo Rcpp, RcppArmadillo

URL `https://smartdata-analysis-and-statistics.github.io/SimTOST/`

NeedsCompilation yes

Author Thomas Debray [aut, cre]
(<https://orcid.org/0000-0002-1790-2719>), Johanna Munoz [aut],
Dewi Amaliah [ctb], Wei Wei [ctb], Marian Mitroiu [ctb], Scott
McDonald [ctb], Biogen Inc [cph, fnd]

Maintainer Thomas Debray <tdebray@fromdatatowisdom.com>

Repository CRAN

Date/Publication 2025-02-18 23:40:08 UTC

Contents

---

get_par *Parameter Configuration for Endpoints and Comparators*

Description

Constructs and returns a list of key parameters (mean vectors, variance-covariance matrices, and allocation rates) required for input into the `sampleSize` function. This function ensures that the parameters for each endpoint and comparator are consistent, properly named, and formatted.

Usage

```
get_par(
  mu_list,
  varcov_list,
  TAR_list,
  type_y = NA,
  arm_names = NA,
  y_names = NA
)
```

Arguments

`mu_list` A list of mean (μ) vectors. Each element in the list represents a comparator, with the corresponding μ vector having a length equal to the number of endpoints.

`varcov_list` A list of variance-covariance matrices. Each element corresponds to a comparator, with a matrix of size $(n \times n)$, where n is the number of endpoints.

`TAR_list` A list of treatment allocation rates (TARs) for each comparator. Each element contains a numeric value (can be fractional or integer) representing the allocation rate for the respective comparator.

`type_y` A numeric vector specifying the type of each endpoint. Use `1` for primary endpoints and `2` for secondary or other endpoints.

`arm_names` (Optional) A character vector containing names of the arms. If not provided, default names (e.g., T1, T2, ...) will be generated.

`y_names` (Optional) A character vector containing names of the endpoints. If not provided, default names (e.g., y1, y2, ...) will be generated.

Value

A named list with the following components:

`mu` A list of mean vectors, named according to `arm_names`.

`varcov` A list of variance-covariance matrices, named according to `arm_names`.

`tar` A list of treatment allocation rates (TARs), named according to `arm_names`.

`type_y` A vector specifying the type of each endpoint.

`weight_seq` A weight sequence calculated from `type_y`, used for endpoint weighting.

`y_names` A vector of names for the endpoints, named as per `y_names`.

#' @details This function ensures that all input parameters (`mu_list`, `varcov_list`, and `TAR_list`) are consistent across comparators and endpoints. It performs checks for positive semi-definiteness of variance-covariance matrices and automatically assigns default names for arms and endpoints if not provided.

Examples

```
mu_list <- list(c(0.1, 0.2), c(0.15, 0.25))
varcov_list <- list(matrix(c(1, 0.5, 0.5, 1), ncol = 2), matrix(c(1, 0.3, 0.3, 1), ncol = 2))
TAR_list <- list(0.5, 0.5)
get_par(mu_list, varcov_list, TAR_list, type_y = c(1, 2), arm_names = c("Arm1", "Arm2"))
```

plot.simss *Plot Power vs Sample Size for Simulation Results*

Description

Generates a detailed plot showing the relationship between power and total sample size for each comparator and the overall combined comparators. The plot also includes confidence intervals for power estimates and highlights the target power with a dashed line for easy visual comparison.

Usage

```
## S3 method for class 'simss'
plot(x, ...)
```

Arguments

x	An object of class simss containing simulation results.
...	Additional arguments to be passed to the plot.simss function for customization.

Details

The plot dynamically adjusts to exclude unnecessary components, such as redundant endpoints or comparators with insufficient data, ensuring clarity and simplicity. The ggplot2 framework is used for visualizations, allowing further customization if needed.

Value

A ggplot object illustrating:

- Power (y-axis) vs. Total Sample Size (x-axis) for individual endpoints and comparators.
- Error bars representing the 95% confidence interval of the power estimates.
- A dashed horizontal line indicating the target power for comparison.
- Faceted panels for each comparator, making it easy to compare results across different groups.

Author(s)

Johanna Muñoz <johanna.munoz@fromdatatowisdom.com>

power_dom *Power Calculation for Difference of Means (DOM) Hypothesis Test*

Description

Computes the statistical power for testing the difference of means (DOM) between two groups using Monte Carlo simulations. The power is estimated based on specified sample sizes, means, standard deviations, and significance level.

Usage

```
power_dom(
  seed,
  mu_test,
  mu_control,
  sigma_test,
  sigma_control,
  N_test,
  N_control,
  lb,
  ub,
  alpha = 0.05,
  nsim = 10000
)
```

Arguments

`seed`	Integer. Seed for reproducibility.
`mu_test`	Numeric. Mean of the test group.
`mu_control`	Numeric. Mean of the control group.
`sigma_test`	Numeric. Standard deviation of the test group.
`sigma_control`	Numeric. Standard deviation of the control group.
`N_test`	Integer. Sample size of the test group.
`N_control`	Integer. Sample size of the control group.
`lb`	Numeric. Lower bound for the equivalence margin.
`ub`	Numeric. Upper bound for the equivalence margin.
`alpha`	Numeric. Significance level (default = 0.05).
`nsim`	Integer. Number of simulations (default = 10,000).

Value

Numeric. Estimated power (probability between 0 and 1).

print.simss *Print Summary of Sample Size Estimation*

Description

Prints the summary results of the sample size estimation for bioequivalence trials, including achieved power, total sample size, and power confidence intervals. The function also details the study design, primary endpoint comparisons, and applied multiplicity corrections.

Usage

```
## S3 method for class 'simss'
print(x, ...)
```

Arguments

`x`	An object of class `"simss"`, typically generated by a sample size estimation function.
`...`	Optional arguments to be passed from or to other methods.

Details

This function displays key metrics from a sample size estimation analysis. It provides an overview of the study design, treatment comparisons, tested endpoints, significance level adjustments, and estimated sample size. For studies with multiple primary endpoints, it describes the multiplicity correction applied.

Value

No return value, called for side effects. The function prints the summary results of the sample size estimation to the console in a structured format.

Author(s)

Thomas Debray <tdebray@fromdatatowisdom.com>

run_simulations_2x2_dom *Run Simulations for a 2x2 Crossover Design with Difference of Means (DOM) test*

Description

This function simulates a 2x2 crossover trial across multiple iterations. It evaluates equivalence across multiple endpoints using the Difference of Means (DOM) test.

Usage

```
run_simulations_2x2_dom(
  nsim,
  n,
  muT,
  muR,
  SigmaW,
  lequi_tol,
  uequi_tol,
  alpha,
  sigmaB,
  dropout,
  Eper,
  Eco,
  typey,
  adseq,
  k,
  arm_seed
)
```

Arguments

`nsim` Integer. The number of simulations to run.

`n` Integer. The sample size per period.

`muT` Numeric vector. Mean outcomes for the active treatment.

`muR` Numeric vector. Mean outcomes for the reference treatment.

`SigmaW` Numeric matrix. Within-subject covariance matrix for endpoints.

`lequi_tol` Numeric vector. Lower equivalence thresholds for each endpoint.

`uequi_tol` Numeric vector. Upper equivalence thresholds for each endpoint.

`alpha` Numeric vector. Significance levels for hypothesis testing across endpoints.

`sigmaB` Numeric. Between-subject variance for the crossover model.

`dropout` Numeric vector of size 2. Dropout rates for each sequence.

`Eper` Numeric vector. Expected period effects for each sequence.

`Eco` Numeric vector. Expected carryover effects for each sequence.

`typey` Integer vector indicating the classification of each endpoint, where `1` corresponds to a primary endpoint and `2` corresponds to a secondary endpoint.

`adseq` Logical. If `TRUE`, applies sequential (hierarchical) testing.

`k` Integer. Minimum number of endpoints required for equivalence.

`arm_seed` Integer vector. Random seed for each simulation.

Details

This function evaluates equivalence using the Difference of Means (DOM) test. Equivalence is determined based on predefined lower (`lequi_tol`) and upper (`uequi_tol`) equivalence thresholds, and hypothesis testing is conducted at the specified significance level (`alpha`). If `adseq` is `TRUE`, primary endpoints must establish equivalence before secondary endpoints are evaluated. The sample size per period is adjusted based on dropout rates, ensuring valid study conclusions. The simulation incorporates within-subject correlation using `SigmaW` and accounts for between-subject variance with `sigmaB`. Expected period effects (`Eper`) and carryover effects (`Eco`) are included in the model. A fixed random seed (`arm_seed`) is used to ensure reproducibility across simulations.

Value

A numeric matrix where each column stores simulation results: The first row (`totaly`) represents the overall equivalence decision (1 = success, 0 = failure). Subsequent rows contain equivalence decisions per endpoint, mean estimates for the treatment group, mean estimates for the reference group, standard deviations for treatment, and standard deviations for reference.

Author(s)

Thomas Debray `<tdebray@fromdatatowisdom.com>`

`run_simulations_2x2_rom` *Run Simulations for a 2x2 Crossover Design with Ratio of Means (ROM) test*

Description

This function simulates a 2x2 crossover trial across multiple iterations. It evaluates equivalence across multiple endpoints using the Ratio of Means (ROM) test.

Usage

```
run_simulations_2x2_rom(
  nsim,
  n,
  muT,
  muR,
  SigmaW,
  lequi_tol,
  uequi_tol,
  alpha,
  sigmaB,
  dropout,
  Eper,
  Eco,
  typey,
  adseq,
  k,
  arm_seed
)
```

Arguments

nsim	Integer. The number of simulations to run.
n	Integer. The sample size per period.
muT	Numeric vector. Mean outcomes for the active treatment.
muR	Numeric vector. Mean outcomes for the reference treatment.
SigmaW	Numeric matrix. Within-subject covariance matrix for endpoints.
lequi_tol	Numeric vector. Lower equivalence thresholds for each endpoint.
uequi_tol	Numeric vector. Upper equivalence thresholds for each endpoint.
alpha	Numeric vector. Significance levels for hypothesis testing across endpoints.
sigmaB	Numeric. Between-subject variance for the crossover model.
dropout	Numeric vector of size 2. Dropout rates for each sequence.
Eper	Numeric vector. Expected period effects for each sequence.
Eco	Numeric vector. Expected carryover effects for each sequence.
typey	Integer vector indicating the classification of each endpoint, where 1 corresponds to a primary endpoint and 2 corresponds to a secondary endpoint.
adseq	Logical. If TRUE, applies sequential (hierarchical) testing.
k	Integer. Minimum number of endpoints required for equivalence.
arm_seed	Integer vector. Random seed for each simulation.

Details

This function evaluates equivalence using the Ratio of Means (ROM) test. Equivalence is determined based on predefined lower lequi_tol and upper uequi_tol equivalence thresholds, and hypothesis testing is conducted at the specified significance level alpha. If adseq is TRUE, primary endpoints must establish equivalence before secondary endpoints are evaluated. The sample size per period is adjusted based on dropout rates, ensuring valid study conclusions. The simulation incorporates within-subject correlation using SigmaW and accounts for between-subject variance with sigmaB. Expected period effects Eper and carryover effects Eco are included in the model. A fixed random seed arm_seed is used to ensure reproducibility across simulations.//'

Value

A numeric matrix where each column stores simulation results: The first row (totaly) represents the overall equivalence decision (1 = success, 0 = failure). Subsequent rows contain equivalence decisions per endpoint, mean estimates for the treatment group, mean estimates for the reference group, standard deviations for treatment, and standard deviations for reference.

@author Thomas Debray <tdebray@fromdatatowisdom.com>

`run_simulations_par_dom` *Run Simulations for a Parallel Design with Difference of Means (DOM) test*

Description

This function simulates a parallel-group trial across multiple iterations. It evaluates equivalence across multiple endpoints using the Difference of Means (DOM) test.

Usage

```
run_simulations_par_dom(
  nsim,
  n,
  muT,
  muR,
  SigmaT,
  SigmaR,
  lequi_tol,
  uequi_tol,
  alpha,
  dropout,
  typey,
  adseq,
  k,
  arm_seed_T,
  arm_seed_R,
  TART,
  TARR,
  vareq
)
```

Arguments

`nsim`	Integer. The number of simulations to run.
`n`	Integer. The sample size per arm (before dropout).
`muT`	arma::vec. Mean vector for the treatment arm.
`muR`	arma::vec. Mean vector for the reference arm.
`SigmaT`	arma::mat. Covariance matrix for the treatment arm.
`SigmaR`	arma::mat. Covariance matrix for the reference arm.
`lequi_tol`	arma::rowvec. Lower equivalence thresholds for each endpoint.
`uequi_tol`	arma::rowvec. Upper equivalence thresholds for each endpoint.
`alpha`	arma::rowvec. Significance level for each endpoint.

`dropout` arma::vec. Dropout rates for each arm (T, R).

`typey` Integer vector indicating the classification of each endpoint, where `1` corresponds to a primary endpoint and `2` corresponds to a secondary endpoint.

`adseq` Boolean. If `TRUE`, applies sequential (hierarchical) testing.

`k` Integer. Minimum number of endpoints required for equivalence.

`arm_seed_T` arma::ivec. Random seed vector for the treatment group (one per simulation).

`arm_seed_R` arma::ivec. Random seed vector for the reference group (one per simulation).

`TART` Double. Treatment allocation ratio (proportion of subjects in treatment arm).

`TARR` Double. Reference allocation ratio (proportion of subjects in reference arm).

`vareq` Boolean. If `TRUE`, assumes equal variances across treatment and reference groups.

Details

Equivalence testing uses either the Difference of Means (DOM) test, applying predefined equivalence thresholds and significance levels. When hierarchical testing (`adseq`) is enabled, all primary endpoints must demonstrate equivalence before secondary endpoints are evaluated. Dropout rates are incorporated into the sample size calculation to ensure proper adjustment. Randomization is controlled through separate random seeds for the treatment and reference groups, enhancing reproducibility.

Value

The function returns an arma::mat storing simulation results row-wise for consistency with R's output format. The first row (`totaly`) contains the overall equivalence decision (1 for success, 0 for failure). The subsequent rows include equivalence deicisons for each endpoint, mean estimates for both treatment and reference groups, and corresponding standard deviations.

Author(s)

Thomas Debray `<tdebray@fromdatatowisdom.com>`

`run_simulations_par_rom` *Run Simulations for a Parallel Design with Ratio of Means (ROM) test*

Description

This function simulates a parallel-group trial across multiple iterations. It evaluates equivalence across multiple endpoints using the Ratio of Means (ROM) test.

Usage

```
run_simulations_par_rom(
  nsim,
  n,
  muT,
  muR,
  SigmaT,
  SigmaR,
  lequi_tol,
  uequi_tol,
  alpha,
  dropout,
  typey,
  adseq,
  k,
  arm_seed_T,
  arm_seed_R,
  TART,
  TARR,
  vareq
)
```

Arguments

`nsim` Integer. The number of simulations to run.

`n` Integer. The sample size per arm (before dropout).

`muT` arma::vec. Mean vector for the treatment arm.

`muR` arma::vec. Mean vector for the reference arm.

`SigmaT` arma::mat. Covariance matrix for the treatment arm.

`SigmaR` arma::mat. Covariance matrix for the reference arm.

`lequi_tol` arma::rowvec. Lower equivalence thresholds for each endpoint.

`uequi_tol` arma::rowvec. Upper equivalence thresholds for each endpoint.

`alpha` arma::rowvec. Significance level for each endpoint.

`dropout` arma::vec. Dropout rates for each arm (T, R).

`typey` Integer vector indicating the classification of each endpoint, where `1` corresponds to a primary endpoint and `2` corresponds to a secondary endpoint.

`adseq` Boolean. If `TRUE`, applies sequential (hierarchical) testing.

`k` Integer. Minimum number of endpoints required for equivalence.

`arm_seed_T` arma::ivec. Random seed vector for the treatment group (one per simulation).

`arm_seed_R` arma::ivec. Random seed vector for the reference group (one per simulation).

`TART` Double. Treatment allocation ratio (proportion of subjects in treatment arm).

`TARR` Double. Reference allocation ratio (proportion of subjects in reference arm).

`vareq` Boolean. If `TRUE`, assumes equal variances across treatment and reference groups.

Details

Equivalence testing uses either the Ratio of Means (ROM) test, applying predefined equivalence thresholds and significance levels. When hierarchical testing (`adseq`) is enabled, all primary endpoints must demonstrate equivalence before secondary endpoints are evaluated. Dropout rates are incorporated into the sample size calculation to ensure proper adjustment. Randomization is controlled through separate random seeds for the treatment and reference groups, enhancing reproducibility.

Value

The function returns an arma::mat storing simulation results row-wise for consistency with R's output format. The first row (`totaly`) contains the overall equivalence decision (1 for success, 0 for failure). The subsequent rows include equivalence decisions for each endpoint, mean estimates for both treatment and reference groups, and corresponding standard deviations.

Author(s)

Thomas Debray <tdebray@fromdatatowisdom.com>

`sampleSize` *Sample Size Calculation for Bioequivalence and Multi-Endpoint Studies*

Description

Computes the required sample size to achieve a target power in studies with multiple endpoints and treatment arms. The function employs modified root-finding algorithms to estimate sample size while accounting for correlation structures, variance assumptions, and equivalence bounds across endpoints. It is particularly useful for bioequivalence trials and multi-arm studies with complex endpoint structures.

Usage

```
sampleSize(
  mu_list,
  varcov_list = NA,
  sigma_list = NA,
  cor_mat = NA,
  sigmaB = NA,
  Eper,
  Eco,
  rho = 0,
  TAR = rep(1, length(mu_list)),
  arm_names = NA,
  ynames_list = NA,
  type_y = NA,
  list_comparator = NA,
  list_y_comparator = NA,
  power = 0.8,
  alpha = 0.05,
  lequi.tol = NA,
  uequi.tol = NA,
  list_lequi.tol = NA,
  list_uequi.tol = NA,
  dtype = "parallel",
  ctype = "ROM",
  vareq = TRUE,
  lognorm = TRUE,
  k = NA,
  adjust = "no",
  dropout = NA,
  nsim = 5000,
  seed = 1234,
  ncores = 1,
  optimization_method = "fast",
  lower = 2,
  upper = 500,
  step.power = 6,
  step.up = TRUE,
  pos.side = FALSE,
  maxiter = 1000,
  verbose = FALSE
)
```

Arguments

mu_list — Named list of arithmetic means per treatment arm. Each element is a vector representing expected outcomes for all endpoints in that arm.

varcov_list — List of variance-covariance matrices, where each element corresponds to a comparator. Each matrix has dimensions: number of endpoints × number of endpoints.

sigma_list — List of standard deviation vectors, where each element corresponds to a comparator and contains one standard deviation per endpoint.

cor_mat — Matrix specifying the correlation structure between endpoints, used along with sigma_list to calculate varcov_list if not provided.

sigmaB — Numeric. Between-subject variance for a 2×2 crossover design.

Eper — Optional numeric vector of length 2 specifying the period effect in a dtype = "2x2" design, applied as c(Period 0, Period 1). Defaults to c(0, 0). Ignored for dtype = "parallel".

Eco — Optional numeric vector of length 2 specifying the carry-over effect per arm in a dtype = "2x2" design, applied as c(Reference, Treatment). Defaults to c(0, 0). Ignored for dtype = "parallel".

rho — Numeric. Correlation parameter applied uniformly across all endpoint pairs. Used with sigma_list to compute varcov_list when cor_mat or varcov_list are not provided.

`TAR` Numeric vector specifying treatment allocation rates per arm. The order must match `arm_names`. Defaults to equal allocation across arms if not provided.

`arm_names` Optional character vector of treatment names. If not supplied, names are derived from `mu_list`.

`ynames_list` Optional list of vectors specifying endpoint names per arm. If names are missing, arbitrary names are assigned based on order.

`type_y` Integer vector indicating endpoint types: `1` for co-primary endpoints, `2` for secondary endpoints.

`list_comparator`
List of comparators. Each element is a vector of length 2 specifying the treatment names being compared.

`list_y_comparator`
List of endpoint sets per comparator. Each element is a vector containing endpoint names to compare. If not provided, all endpoints common to both comparator arms are used.

`power` Numeric. Target power (default = 0.8).

`alpha` Numeric. Significance level (default = 0.05).

`lequi.tol` Numeric. Lower equivalence bounds (e.g., -0.5) applied uniformly across all endpoints and comparators.

`uequi.tol` Numeric. Upper equivalence bounds (e.g., 0.5) applied uniformly across all endpoints and comparators.

`list_lequi.tol` List of numeric vectors specifying lower equivalence bounds per comparator.

`list_uequi.tol` List of numeric vectors specifying upper equivalence bounds per comparator.

`dtype` Character. Trial design: `"parallel"` (default) for parallel-group or `"2x2"` for crossover (only for 2-arm studies).

`ctype` Character. Hypothesis test type: `"DOM"` (Difference of Means) or `"ROM"` (Ratio of Means).

`vareq` Logical. Assumes equal variances across arms if `TRUE` (default = `FALSE`).

`lognorm` Logical. Whether data follows a log-normal distribution (`TRUE` or `FALSE`).

`k` Integer vector. Minimum number of successful endpoints required for global bioequivalence per comparator. Defaults to all endpoints per comparator.

`adjust` Character. Alpha adjustment method: `"k"` (K-fold), `"bon"` (Bonferroni), `"sid"` (Sidak), `"no"` (default, no adjustment), or `"seq"` (sequential).

`dropout` Numeric vector specifying dropout proportion per arm.

`nsim` Integer. Number of simulated studies (default = 5000).

`seed` Integer. Seed for reproducibility.

`ncores` Integer. Number of processing cores for parallel computation. Defaults to `1`. Set to `NA` for automatic detection (`ncores - 1`).

`optimization_method`
Character. Sample size optimization method: `"fast"` (default, root-finding algorithm) or `"step-by-step"`.

lower Integer. Minimum sample size for search range (default = 2).

upper Integer. Maximum sample size for search range (default = 500).

step.power Numeric. Initial step size for sample size search, defined as 2^step.power. Used when optimization_method = "fast".

step.up Logical. If TRUE (default), search increments upward from lower; if FALSE, decrements downward from upper. Used when optimization_method = "fast".

pos.side Logical. If TRUE, finds the smallest integer i closest to the root such that f(i) > 0. Used when optimization_method = "fast".

maxiter Integer. Maximum iterations allowed for sample size estimation (default = 1000). Used when optimization_method = "fast".

verbose Logical. If TRUE, prints progress and messages during execution (default = FALSE).

Value

A list containing:

response Array summarizing simulation results, including estimated sample sizes, achieved power, and confidence intervals.

table.iter Data frame showing estimated sample sizes and calculated power at each iteration.

table.test Data frame containing test results for all simulated trials.

param.u Original input parameters.

param Final adjusted parameters used in sample size calculation.

param.d Trial design parameters used in the simulation.

Author(s)

Johanna Muñoz <johanna.munoz@fromdatatowisdom.com>

References

Schuirmann, D. J. (1987). A comparison of the Two One-Sided Tests procedure and the Power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657-680. doi:10.1007/BF01068419

Mielke, J., Jones, B., Jilma, B., & König, F. (2018). Sample size for multiple hypothesis testing in biosimilar development. *Statistics in Biopharmaceutical Research, 10*(1), 39-49. doi:10.1080/19466315.2017.1371071

Berger, R. L., & Hsu, J. C. (1996). Bioequivalence trials, intersection-union tests, and equivalence confidence sets. *Statistical Science*, 283-302.

Sozu, T., Sugimoto, T., Hamasaki, T., & Evans, S. R. (2015). "Sample Size Determination in Clinical Trials with Multiple Endpoints." *SpringerBriefs in Statistics*. doi:10.1007/978-3-319-22005-5

Examples

```
mu_list <- list(SB2 = c(AUCinf = 38703, AUClast = 36862, Cmax = 127.0),
                EUREF = c(AUCinf = 39360, AUClast = 37022, Cmax = 126.2),
                USREF = c(AUCinf = 39270, AUClast = 37368, Cmax = 129.2))

sigma_list <- list(SB2 = c(AUCinf = 11114, AUClast = 9133, Cmax = 16.9),
                   EUREF = c(AUCinf = 12332, AUClast = 9398, Cmax = 17.9),
                   USREF = c(AUCinf = 10064, AUClast = 8332, Cmax = 18.8))

# Equivalent boundaries
lequi.tol <- c(AUCinf = 0.8, AUClast = 0.8, Cmax = 0.8)
uequi.tol <- c(AUCinf = 1.25, AUClast = 1.25, Cmax = 1.25)

# Arms to be compared
list_comparator <- list(EMA = c("SB2", "EUREF"),
                        FDA = c("SB2", "USREF"))

# Endpoints to be compared
list_y_comparator <- list(EMA = c("AUCinf", "Cmax"),
                          FDA = c("AUClast", "Cmax"))

# Equivalence boundaries for each comparison
lequi_lower <- c(AUCinf = 0.80, AUClast = 0.80, Cmax = 0.80)
lequi_upper <- c(AUCinf = 1.25, AUClast = 1.25, Cmax = 1.25)

# Run the simulation
sampleSize(power = 0.9, alpha = 0.05, mu_list = mu_list,
           sigma_list = sigma_list, list_comparator = list_comparator,
           list_y_comparator = list_y_comparator,
           list_lequi.tol = list("EMA" = lequi_lower, "FDA" = lequi_lower),
           list_uequi.tol = list("EMA" = lequi_upper, "FDA" = lequi_upper),
           adjust = "no", dtype = "parallel", ctype = "ROM", vareq = FALSE,
           lognorm = TRUE, ncores = 1, nsim = 50, seed = 1234)
```

sampleSize_Mielke *Sample Size Estimation for Multiple Hypothesis Testing Using Mielke's Method*

Description

Estimates the required sample size to achieve a specified power level for multiple hypothesis testing, using the approach described by Mielke et al. (2018). This function is particularly useful for bioequivalence or biosimilar studies with multiple correlated endpoints, where a minimum number of endpoints must meet equivalence criteria.

Usage

```
sampleSize_Mielke(
  power,
  Nmax,
  m,
  k,
  rho,
  sigma,
  true.diff,
  equi.tol,
  design,
  alpha,
  adjust = "no",
  seed = NULL,
  nsim = 10000
)
```

Arguments

`power` Numeric. Desired statistical power.

`Nmax` Integer. Maximum allowable sample size.

`m` Integer. Total number of endpoints.

`k` Integer. Number of endpoints that must meet the success criteria for overall study success.

`rho` Numeric. Constant correlation coefficient among endpoints.

`sigma` Numeric or vector. Standard deviation of each endpoint. If a single value is provided, it is assumed to be constant across all endpoints. In a 2x2 crossover design, this is the within-subject standard deviation; in a parallel design, it represents the treatment group's standard deviation, assumed to be the same for both test and reference.

`true.diff` Numeric or vector. Assumed true difference between test and reference for each endpoint. If a single value is provided, it is applied uniformly across all endpoints.

`equi.tol` Numeric. Equivalence margin; the equivalence interval is defined as (-equi.tol, +equi.tol).

`design` Character. Study design, either "22co" for a 2x2 crossover design or "parallel" for a parallel groups design.

`alpha` Numeric. Significance level for the hypothesis test.

`adjust` Character. Method for multiplicity adjustment; options are "no" (no adjustment), "bon" (Bonferroni adjustment), and "k" (k-adjustment).

`seed` Integer. Random seed for reproducibility.

`nsim` Integer. Number of simulations to run for power estimation (default: 10,000).

Details

This function uses the method proposed by Mielke et al. (2018) to estimate the sample size required to achieve the desired power level in studies with multiple correlated endpoints. The function iteratively increases sample size until the target power is reached or the maximum allowable sample size (Nmax) is exceeded. The approach accounts for endpoint correlation and supports adjustments for multiple testing using various correction methods.

Value

A named vector containing:

"power.a" Achieved power with the estimated sample size.

"SS" Required sample size per sequence to achieve the target power.

References

Mielke, J., Jones, B., Jilma, B. & König, F. Sample Size for Multiple Hypothesis Testing in Biosimilar Development. Statistics in Biopharmaceutical Research 10, 39–49 (2018).

Examples

```
# Example 1 from Mielke
sampleSize_Mielke(power = 0.8, Nmax = 1000, m = 5, k = 5, rho = 0,
                  sigma = 0.3, true.diff =  log(1.05), equi.tol = log(1.25),
                  design = "parallel", alpha = 0.05, adjust = "no",
                  seed = 1234, nsim = 100)
```

simParallelEndpoints *Generate Simulated Endpoint Data for Parallel Group Design*

Description

Generate simulated endpoint data for a parallel design, with options for normal and lognormal distributions.

Usage

```
simParallelEndpoints(
  n,
  mu.arithmetic,
  mu.geometric = NULL,
  Sigma,
  CV = NULL,
  seed,
  dist = "normal"
)
```

Arguments

n	Integer. The sample size for the generated data.
mu.arithmetic	Numeric vector. The arithmetic mean of the endpoints on the original scale.
mu.geometric	Numeric vector. The geometric mean of the endpoints on the original scale. Only used if dist = "lognormal".

`Sigma`	Matrix. Variance-covariance matrix of the raw data on the original scale. If `dist = "lognormal"`, this matrix is transformed to the log scale.
`CV`	Numeric vector. Coefficient of variation (CV) of the raw data. Only used when `dist = "lognormal"`, where it is transformed to the log scale.
`seed`	Integer. Seed for random number generation, ensuring reproducibility.
`dist`	Character. Assumed distribution of the endpoints: either `"normal"` or `"lognormal"`.

Value

A matrix of simulated endpoint values for a parallel design, with dimensions `n` by the number of variables in `mu.arithmetic` or `mu.geometric`.

Author(s)

Thomas Debray <`tdebray@fromdatatowisdom.com`>

`SimTOST` *Sample Size Estimation via Simulation*

Description

SimTOST: A Package for Sample Size Simulations

Details

The SimTOST package provides tools for simulating sample sizes, calculating power, and assessing type-I error for various statistical scenarios.

Author(s)

Maintainer: Thomas Debray <`tdebray@fromdatatowisdom.com`> (ORCID)

Authors:

- Johanna Munoz <`johanna.munoz@fromdatatowisdom.com`>

Other contributors:

- Dewi Amaliah <`dewi.amaliah@fromdatatowisdom.com`> [contributor]
- Wei Wei <`wei.wei@biogen.com`> [contributor]
- Marian Mitroiu <`marian.mitroiu@biogen.com`> [contributor]
- Scott McDonald <`scott.mcdonald@fromdatatowisdom.com`> [contributor]
- Biogen Inc [copyright holder, funder]

References

Mielke, J., Jones, B., Jilma, B. & König, F. Sample Size for Multiple Hypothesis Testing in Biosimilar Development. *Statistics in Biopharmaceutical Research* 10, 39–49 (2018).

See Also

sampleSize

test_2x2_dom *Simulate a 2x2 Crossover Design and Compute Difference of Means (DOM)*

Description

Simulates a two-sequence, two-period (2x2) crossover design and evaluate equivalence for the difference of means (DOM).

Usage

```
test_2x2_dom(
  n,
  muT,
  muR,
  SigmaW,
  lequi_tol,
  uequi_tol,
  alpha,
  sigmaB,
  dropout,
  Eper,
  Eco,
  typey,
  adseq,
  k,
  arm_seed
)
```

Arguments

`n`	integer number of subjects per sequence
`muT`	vector mean of endpoints on treatment arm
`muR`	vector mean of endpoints on reference arm
`SigmaW`	matrix within subject covar-variance matrix across endpoints
`lequi_tol`	vector lower equivalence tolerance band across endpoints
`uequi_tol`	vector upper equivalence tolerance band across endpoints
`alpha`	vector alpha value across endpoints
`sigmaB`	double between subject variance (assumed same for all endpoints)
`dropout`	vector of size 2 with dropout proportion per sequence (0,1)
`Eper`	vector of size 2 with period effect on period (0,1)

Eco	vector of size 2 with carry over effect of arm c(Reference, Treatment).
typey	vector with positions of primary endpoints
adseq	boolean is used a sequential adjustment?
k	integer minimum number of equivalent endpoints
arm_seed	seed for the simulation

Value

A numeric matrix containing the simulated hypothesis test results. The first column represents the overall equivalence decision, where 1 indicates success and 0 indicates failure. The subsequent columns contain the hypothesis test results for each endpoint, followed by mean estimates for the reference and treatment groups, and standard deviations for the reference and treatment groups.

test_2x2_rom *Simulate a 2x2 Crossover Design and Compute Ratio of Means (ROM)*

Description

Simulates a two-sequence, two-period (2x2) crossover design and evaluate equivalence for the ratio of means (ROM).

Usage

```
test_2x2_rom(
  n,
  muT,
  muR,
  SigmaW,
  lequi_tol,
  uequi_tol,
  alpha,
  sigmaB,
  dropout,
  Eper,
  Eco,
  typey,
  adseq,
  k,
  arm_seed
)
```

Arguments

n	integer number of subjects per sequence
muT	vector mean of endpoints on treatment arm

`muR`	vector mean of endpoints on reference arm
`SigmaW`	matrix within subject covar-variance matrix across endpoints
`lequi_tol`	vector lower equivalence tolerance band across endpoints
`uequi_tol`	vector upper equivalence tolerance band across endpoints
`alpha`	vector alpha value across endpoints
`sigmaB`	double between subject variance (assumed same for all endpoints)
`dropout`	vector of size 2 with dropout proportion per sequence (0,1)
`Eper`	vector of size 2 with period effect on period (0,1)
`Eco`	vector of size 2 with carry over effect of arm c(Reference, Treatment).
`typey`	vector with positions of primary endpoints
`adseq`	boolean is used a sequential adjustment?
`k`	integer minimum number of equivalent endpoints
`arm_seed`	seed for the simulation

Value

A numeric matrix containing the simulated hypothesis test results. The first column represents the overall equivalence decision, where 1 indicates success and 0 indicates failure. The subsequent columns contain the hypothesis test results for each endpoint, followed by mean estimates for the reference and treatment groups, and standard deviations for the reference and treatment groups.

`test_par_dom` *Simulate a Parallel Design and Test Difference of Means (DOM)*

Description

Simulates a parallel-group design and performs equivalence testing using the difference of means (DOM) approach. This function evaluates whether the treatment and reference groups are equivalent based on predefined equivalence margins and hypothesis testing criteria.

Usage

```
test_par_dom(
  n,
  muT,
  muR,
  SigmaT,
  SigmaR,
  lequi_tol,
  uequi_tol,
  alpha,
  dropout,
  typey,
  adseq,
  k,
  arm_seedT,
  arm_seedR,
  TART,
  TARR,
  vareq
)
```

Arguments

n	integer number of subjects per arm
muT	vector mean of endpoints on treatment arm
muR	vector mean of endpoints on reference arm
SigmaT	matrix covar-variance matrix on treatment arm across endpoints
SigmaR	matrix covar-variance matrix on reference arm across endpoints
lequi_tol	vector lower equivalence tolerance band across endpoints
uequi_tol	vector upper equivalence tolerance band across endpoints
alpha	vector alpha value across endpoints
dropout	vector of size 2 with dropout proportion per arm (T,R)
typey	vector with positions of primary endpoints
adseq	boolean is used a sequential adjustment?
k	integer minimum number of equivalent endpoints
arm_seedT	integer seed for the simulation on treatment arm
arm_seedR	integer seed for the simulation on reference arm
TART	double treatment allocation rate for the treatment arm
TARR	double treatment allocation rate for the reference arm
vareq	boolean assumed equivalence variance between arms for the t-test

Details

The function simulates a parallel-group study design and evaluates equivalence using the difference of means (DOM) approach. It accounts for dropout rates and treatment allocation proportions while generating simulated data based on the specified covariance structure. The test statistics are computed, and a final equivalence decision is made based on the predefined number of required significant endpoints (k). If sequential testing (adseq) is enabled, primary endpoints must establish equivalence before secondary endpoints are evaluated. When vareq = TRUE, the test assumes equal variances between groups and applies Schuirmann's two one-sided tests (TOST).

Value

A numeric matrix containing the simulated hypothesis test results. The first column represents the overall equivalence decision, where 1 indicates success and 0 indicates failure. The subsequent columns contain the hypothesis test results for each endpoint, followed by mean estimates for the reference and treatment groups, and standard deviations for the reference and treatment groups.

`test_par_rom` *Simulate a Parallel Design and Test Ratio of Means (ROM)*

Description

Simulates a parallel-group design and performs equivalence testing using the ratio of means (ROM) approach. This function evaluates whether the treatment and reference groups are equivalent based on predefined equivalence margins and hypothesis testing criteria.

Usage

```
test_par_rom(
  n,
  muT,
  muR,
  SigmaT,
  SigmaR,
  lequi_tol,
  uequi_tol,
  alpha,
  dropout,
  typey,
  adseq,
  k,
  arm_seedT,
  arm_seedR,
  TART,
  TARR,
  vareq
)
```

Arguments

`n`	integer number of subjects per arm
`muT`	vector mean of endpoints on treatment arm
`muR`	vector mean of endpoints on reference arm
`SigmaT`	matrix covar-variance matrix on treatment arm across endpoints
`SigmaR`	matrix covar-variance matrix on reference arm across endpoints
`lequi_tol`	vector lower equivalence tolerance band across endpoints
`uequi_tol`	vector upper equivalence tolerance band across endpoints
`alpha`	vector alpha value across endpoints
`dropout`	vector of size 2 with dropout proportion per arm (T,R)
`typey`	vector with positions of primary endpoints
`adseq`	boolean is used a sequential adjustment?

`k` integer minimum number of equivalent endpoints

`arm_seedT` integer seed for the simulation on treatment arm

`arm_seedR` integer seed for the simulation on reference arm

`TART` double treatment allocation rate for the treatment arm

`TARR` double treatment allocation rate for the reference arm

`vareq` Boolean. If `TRUE`, assumes equal variance between arms and applies Schuirmann's two one-sided tests (TOST) for equivalence using a pooled variance.

Details

The function simulates a parallel-group study design and evaluates equivalence using the ratio of means (ROM) approach. It accounts for dropout rates and treatment allocation proportions while generating simulated data based on the specified covariance structure. The test statistics are computed, and a final equivalence decision is made based on the predefined number of required significant endpoints (`k`). If sequential testing (`adseq`) is enabled, primary endpoints must establish equivalence before secondary endpoints are evaluated. When `vareq = TRUE`, the test assumes equal variances between groups and applies Schuirmann's two one-sided tests (TOST).

Value

A numeric matrix containing the simulated hypothesis test results. The first column represents the overall equivalence decision, where 1 indicates success and 0 indicates failure. The subsequent columns contain the hypothesis test results for each endpoint, followed by mean estimates for the reference and treatment groups, and standard deviations for the reference and treatment groups.

Index